Precision Landing for Drone based on Computer Vision

Petra Matatias Situmorang, Astria Nur Irfansyah, and Muhammad Attamimi
Electrical Engineering Department, Faculty of Electrical Technology, Institut Teknologi Sepuluh Nopember
Jalan Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: irfansyah@ee.its.ac.id

***Abstract*— Unmanned Aerial Vehicle (UAV) or popularly known as drones are already widely used to help human work, for example delivering kits, monitoring and mapping an area (mapping and monitoring) and photography. One ongoing study regarding UAV is the automatic landing of a multi-copter drone. Common navigation methods for this purpose is the GPS navigation system and integrated GPS with visual navigation system. GPS has accuracy limitations, therefore in this work, we apply computer vision to assist quadcopter drone to perform automatic precision landing. The camera will help the vehicle to find the landing pad. The visual system can get the relative position of the vehicle towards the target, which is required for the control system of the UAV to reach the landing point. Both stability and accuracy of the landing are the focus and challenges of this work. A prototype quadcopter with an on-board Raspberry Pi 3B+ and camera, to implement the computer vision algorithm, and an Omnibus flight controller, was constructed and tested, with a landing platform of a specified pattern. The drone landed with an average error of 45 cm with a 21cm minimum error, better than the average landing error of 2.85 m based on GPS only**.

Keyword—image processing, drone, precision landing.

I. INTRODUCTION

Research in Unmanned Aerial Vehicles (UAV), or simply drones, has been gaining interest. There are numerous research on UAV technology by academics and researchers to help humans in their work. Some of the examples are drone for military activity, mapping and monitoring, photography, etc. A common type of UAV takes the form of the quadcopter, with the advantage of vertical take-off and landing (VTOL) capability.

Some of the existing VTOL system includes GPS-based landing system [3], camera-based landing system, and also integrated GPS and visual landing system. The downside of the first landing system, which is GPS-based landing system, is that this system relies completely on GPS sensor which generally has low accuracy and can only be operated outdoor. As for its use for landing, most consumer-grade GPS devices has 8m radius accuracy [4]. Furthermore, most landing system using GPS navigation for quadrotor has an average error of 5 meter[5], which can be considered as a significant error. In comparison, camera suits better for the device used in detecting the landing pad.

Computer vision on drone will help to determine the relative position of the vehicle against the landing pad. One of the advantage of using camera is this sensor is considerably well-balanced because it can be used on both indoor and outdoor. On top of it, using visual system on drone means that drone can achieve more information of the surrounding environment, having a good adaptation, small-sized and light-weight[2]. By integrating both GPS and camera, navigation of the drone can be more accurate. Image processing on the drone takes part to detect the center point of the landing pad, which will be the reference to increase the drone landing accuracy. With this integration system, it is hoped that landing will be more precise, so that this landing system can be used for better usage of drone. In the next section, we review the existing literature on visually aided automatic drone landing and provide a discussion on our design approach and implementation.

II. LITERATURE REVIEW

Unmanned Aerial Vehicle (UAV) or drone is a vehicle that does not need humans to operate them [19]. One of the examples is quadcopter, a drone which has four motors in total, where two of the motors rotate clockwise while other two rotate counter clockwise to stabilize the vehicle's position[19],[10]. Numerous drone applications have helped humans on their work, such as logistic delivery to a dangerous area that cannot be reached by humans or also known as dead zone[11]. Under certain circumstance, the targeted area cannot be reached by only utilizing GPS. Image processing is capable to assist GPS in order to obtain the position of landing pad for the automatic landing[2], [3], [11]–[13].

A. Computer Vision on Drone

Computer vision technology on drone has been well-developed. This technology has been used for navigation system [13], [14], automatic landing [11], avoiding obstacles and collisions, and also used for mapping. Optical flow modules which relies on image processing are now also available as modules that can greatly help drone hold its position on flight.

As being mentioned above, automatic landing based on GPS does not guarantee the drone to land precisely. The error on the landing is considerably big, around the number of 5 meter using typical GPS devices. The solution of this problem is to add another sensor to compensate the poor performance of single GPS sensors, which is using visual sensor, commonly being a camera[11].

Numerous researchers have developed the technology of visual-based drone, for example Venugopalan [3] has conducted a development for automatic landing technology for drone on an autonomous sea vehicle. The algorithm adopted for the experiment consists of two major parts. First, image processing is employed to detect the landing pad. Second, a control design is implemented for the navigation. The landing pad detection method is implemented by combining pattern detection of the landing pad with color-based detection.

In order to maintain a constant altitude for the camera, drone is often equipped with altitude sensor. For this reason, some researchers use ultrasonic sensor like the experiments on [2] and [12], while others make use of barometer sensor [12] , and also LIDAR [15]. With the help of altitude sensor, the prediction of z-coordinate of the drone can be achieved, making up the usage of visual sensor previously to obtain x and y coordinate position of the drone. Using the x, y, and z position's information of the drone, landing system can then be designed.

B. Landing and Control System

In short, landing process of a drone begins with camera taking frames from video, and then processing them to obtain the information of the drone position. Based on this information, the drone can decide how to move to the landing position. The whole process is divided into two parts, which are search routine and landing [3]. Search routine is the process where the drone looks up for the landing pad. At first, drone flies at certain altitude. If drone can't find the landing pad on that position, then it must fly higher as to increase the camera's coverage area, to the point where the camera of the drone finds the landing pad[2]. After that, drone moves to the landing position with pitch and roll control. When the drone has reached the landing position (still in flight), it's when the landing routine takes place. At this point, PID control for pitch and roll is utilized to adjust drone's position to match the center point of the landing pad.

One of the problem which often happens is the landing pad disappears from the camera. This is because the occurrence of drifting, where the drone moves out of control. To solve this problem, control system needs to use the previous data to help the drone returning to its prior position until it is back to where the landing pad is detected. For another note, while drone is descending, the coverage area of the camera is also decreased. If the descend happens too fast, it's possible that the landing pad is no longer detected by the camera. Because of that, while descending, it's necessary to link the throttle with the deviation distance from the target [2]. If the deviation distance is pretty far, drone needs to descend slower, and vice versa – if the deviation distance is close to zero, drone can descend faster.

C. Precision Landing

Automatic landing system becomes an important thing in drone development. One of the examples is the application of quadcopter which is able to perform automatic landing with high accuracy[2][3]. Precision landing on drone is developed to achieve the need of decent-accuracy landing. Precision landing's development often involves the use of computer vision to land on a certain platform with fixed size. One of the research about precision landing is conducted using a low-priced sensor [5]. The research concludes that its designed system is able to perform precision landing with average error of 0.3705m for the indoor experiment and 0.388m for the outdoor experiment.

There is also precision landing using optical flow and integrated GPS/INS[16]. Based on the experiment mentioned, precision landing can be achieved with maximum error only ranges on the number of 0.27m. This experiment is conducted with various altitude, ranging from 60 cm to 150 cm.

Automatic landing system is also developed using computer vision to detect and recognize correct target[17]. The helipad target has the size of 120x120 cm. The system designed combines visual system and GPS system for the navigation. This study achieves precision landing with average position error of 42 cm.

Other experiment is centered on the use of optical flow for the development of automatic UAV navigation [18]. In this experiment, optical flow principle is used to obtain the 2D position of the drone. With the addition of ultrasonic sensor, infrared, and pressure sensor, the experiment concludes that it's able to estimate altitude so that the 3D position of the drone can be achieved. As the result of the system, it's able to perform precision landing with error of 30 cm.

For another experiment, precision landing's development mainly utilizes the concept of computer vision[19]. The researcher mentions that based on the simulation result, the designed system is capable to land precisely with average error of 5 cm, while on the real execution, the average error ranges around 30 cm.

There is another research that focuses on the use of optical flow sensor, but with the addition of adaptive algorithm fuzzy data fusion to obtain the translation value of x and y from the sensor [20]. This algorithm is mainly developed to obtain the accurate estimation of drone's position while on landing process. This method is conducted indoor and achieves precision landing with maximum position error of 10 cm.

Another development of precision landing consists of the use of telemetry data exchange and fusion of Global Navigational Satellite System (GNSS) sensor, infrared marker detection and some other sensors[21]. In this study, researcher uses vision system that is built by marker recognition and infrared camera. This vision system performs better in term of stability because of the usage of infrared marker. With this method, researcher is able to land precisely with position error of 8 cm.

Precision landing can also be developed by combining multiple addition sensor to improve landing accuracy compared to only use GPS/INS. The most popular research is development of landing system using additional visual sensor. These developments are conducted in hope to achieve the precision landing where drone will be able to land on certain platform.

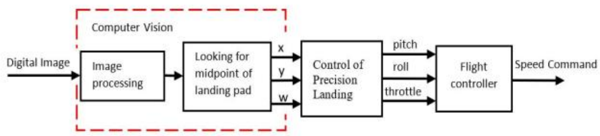

Figure 1. Overall system block diagram

III. SYSTEM DESIGN

In this section, we describe the approaches that this work takes in implementing the precision landing drone.

A. Computer Vision

In this work, the computer vision is used to obtain the coordinates of the relative position of the landing pad against the drone. Figure 1 is overall block diagram of system. This computer vision aims to detect landing pads as in Figure 2.

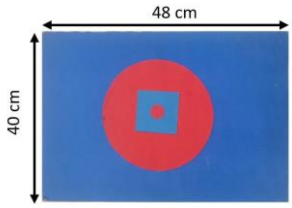

Figure 2. Landing pad

The size of the landing pad to be detected is 48x40 cm. The computer vision algorithm is implemented on a Raspberry Pi.

Before designing an image processing algorithm, the landing pad is designed with certain shapes and colors to be easily detected by the camera. In this work, this is achieved using a small red circle within a blue square, which is drawn inside a larger red circle (Fig. 2). In order to strengthen and facilitate detection, we use the combination of two colors. Generally, image processing is designed to find the midpoint of the landing pad. Starting with loading digital images from the camera. The image loaded by the camera is still in the RGB color space. RGB color space is the most popular type in the real world, but to be processed on computer vision, RGB is very sensitive to light intensity changes so other color spaces must be sought. HSV is one of the most commonly used in computer vision preprocessing.

It can be seen that Figure 3 is a block of image processing for detecting landing pad. The method used is to use a color threshold. The process is done by doing a threshold of two colors. After a threshold of two colors is done, by using the findContour and drawContour functions on the opencv that are applied to the threshold result 1 or outside by setting the thickness parameter to -1, on the cv2.drawContours syntax. If you only use one time threshold, the object detected is very susceptible to noise and many undesirable similar colors are also detected. So by using logic AND between the pixel values on the results of the contour drawing from the first threshold with the result of the second threshold, the right figure 4 is generated. The resulting image will be better. With this method the object that is seen is red which is between blue. So even though there is another red color, it will not be detected by the camera if it is not in blue.

1. Looking for the Midpoint of the Landing Pad

The result of image processing is landing pad detection. After getting the picture it can be processed to find out the position on the camera captured image. Using the findcontour function can easily determine the midpoint of the detected contour. The detected contour will be drawn into a white blob. Then using cv2.boundrect () to get the blob coordinate position detected along with its length and width.

By using cv2.boundRect () syntax there are four parameters, namely the coordinates of the square angles that surround the BLOB, the length and width of BLOB. Value (x, y) is the pixel coordinate of the upper left corner of the square surrounding the blob or contour detected. Then w and h are square lengths and widths. To reduce noise, namely small blobs, it will be filtered with a pixel area of at least 100 pixels. The cv2.contourArea function can get the area of each detected contour.

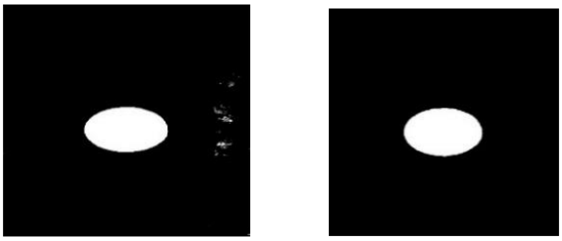
Figure 4. Result of second threshold (left), final result (right)

To find out the distance between the drone and the midpoint of the landing pad is to give a sign at the midpoint of the captured image.

B. Control of Precision Landing

The control system in this work is a simple rule-based control system, using an Arduino microcontroller acting as the high-level controller giving commands in the form of PPM signals in various channels to the Omnibus flight controller of the quadcopter. The controller takes the relative landing pad position from the Raspberry Pi. The flight controller is configured to the position hold flight mode. Therefore, the automatic landing process controlled by the Arduino module will control the pitch, roll and throttle values. The yaw channel remains independent.

When Arduino runs an automatic mode, the data sent by Raspberry Pi is readed through serial communication. There are three accepted values, namely the coordinates of the relative position of the drone to the landing pad (x, y) and the width of the blob (w) threshold result. By utilizing pixel width, we can estimate the height of the drone. The wider the blob detected can be assumed that the drone is getting closer to the landing pad. The higher the drone, the less detected blob will be. When the blob (w) width value data is smaller than 100 pixels, which means the height of the drone is more than 30 cm, the command that is executed is to position it to the landing pad and slowly descend. During the process of centering the position, the value of x, y keeps being stored. So whenever the drone moves past the landing pad until it is no longer visible, the previously stored value will indicate the relative direction of the landing pad, and the drone will move to find this last position. When the width of the blob is smaller than zero, meaning that the landing pad is not detected, the drone is instructed to increase the height to expand the camera's view until the ground is detected again by the camera.

The control system centers on the position of the drone using an approximate drone position algorithm based on the

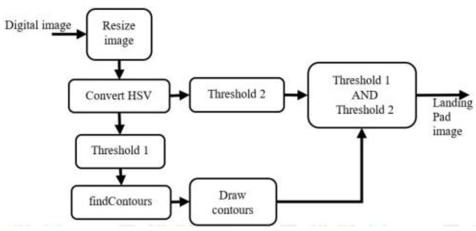

Figure 3. Block diagram of Image processing

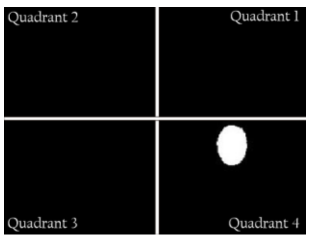

Figure 5. Distribution of camera image quadrants

cartesian quadrant. The image from the camera will be divided into 4 parts. Quadrant 1 is the upper right, quadrant 2 is the upper left, quadrant 3 is in the lower left and quadrant 4 is the lower right. The image size from image processing is 160x160 pixels. This small resolution choice is made for faster processing time and higher frame rates on the on-board Raspberry Pi. The midpoint is at the point (80,80). Based on the camera installation on the drone, point (0,0) pixels are in the upper left corner. It is assumed that the position of the drone is at the midpoint of the image (80,80). Figure 5 shows the division of the quadrants of the camera frame. The white circle is the landing pad detected by the camera.

To control the stable movement of the drone, fixed slow movement commands are given from the Arduino board to the Omnibus flight controller. Arduino output commands are: forward (pitch inclination angle = + 2.1°), backward (inclination angle pitch = -2.1°), left (inclination angle roll = -2.1°) and right (inclination angle roll = +2.1°) . The addition and subtraction of the value is obtained by trial and error until the movement of the drone is sufficiently stable for landing. The y axis is the forward or backward movement of the drone, the x axis is the left and right of the drone. The output value for the control command is the pulse of each channel with a range of values of 1000-2000. The drone is at a standstill when the output pulse value is the middle value of 1500. The range of pitch and roll inclination angle is determined at 0 ° - 30 °. During landing, throttle will be constantly reduced 2.5% every 2 seconds, as long as the landing pad is detected.

C. Drone

Figure 6 shows the overall system design of the device on the drone. During mechanical design, the center of gravity of the drone was made as close as possible to the midpoint of the drone. The weight distribution of each component must be the same. Incorrect placement of components that causes unbalanced weight will cause unstable flight. The challenge is the small size of the frame so that the placement of electronic components is inadequate, so that the mechanics are given additional space for placement of electronic components.

The raspberry pi camera is on the right side of the drone. The camera is mounted facing downwards and not too far from the center of gravity. The position of the camera placement will affect the calculation of the midpoint value of the digital image captured by the camera. The position of raspberry pi camera placement can be seen in figure 7.

Design of the drone electronics firstly consists of

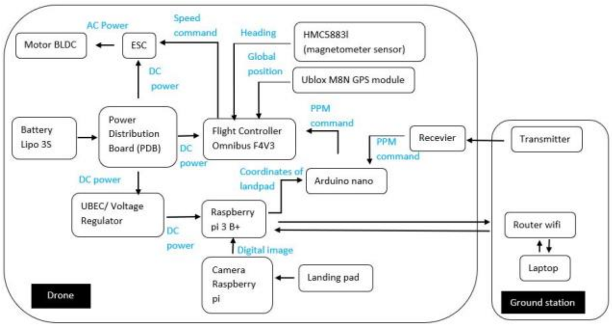

Figure 6. Block diagram of drone hardware

determining the supply of voltage and distribution to each component, as well as the wiring for the connection of each

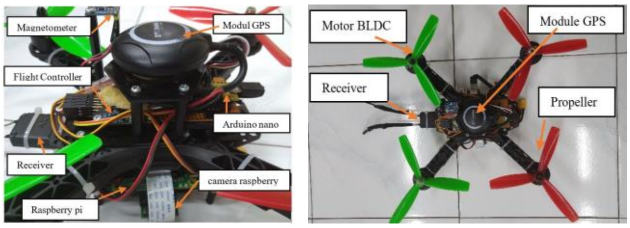

Figure 7. Drone side view (left) top view (right)

component used. The battery used to supply drones is lithium polymer (LiPo) 3 cell battery with 1500mAh capacity. The voltage is distributed to each component through the PDB (power distribution board).

IV. RESULT AND ANALYSIS

A. Flight Mode Test

The first test is flying the drone in angle mode and position hold mode. A more stable flight mode will be very helpful in the landing process. The angle mode is one example of non-navigation mode. Angle flight mode will limit pitch and roll inclination angles. Whereas the position hold flight mode is one of the navigation modes that maintains a 3-dimensional position. Reference point to maintain its position based on GPS sensor readings.

Referring to Fig. 8, the path of the drone in flight mode angle can be seen. Position hold flight mode is shown in Fig. 9. The x axis is the longitude value, the y axis is the latitude value and the z axis is the altitude value. From both figures,

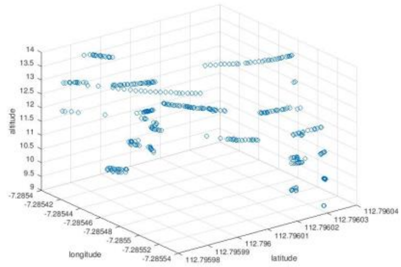

Figure 8. Drone trajectory in flight mode angle

it can be seen that the drone flying in the angle mode has considerably more drifts. In comparison, with the position hold mode, the drone trajectory is more concentrated in one area.

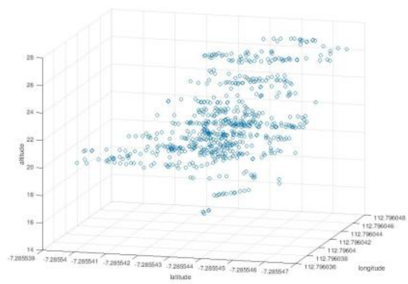

Figure 9. Drone trajectory in flight mode: position hold

Table 1
Landing using GPS

Trial	Error position
1	305 cm
2	100 cm
3	473 cm
4	422 cm
5	528 cm
6	130 cm
7	368 cm
8	26 cm
9	338 cm
10	164 cm

B. Landing Using GPS

Automatic landing using GPS only is performed in this work in order to compare it with the visually-assisted drone landing.

The tests were conducted in the futsal field of Electrical Engineering, Sepuluh Nopember Institute of Technology. This landing test is done by adding 1 waypoint to be addressed by the drone. After reaching the waypoint, the drone will execute the RTH (return to home) command and land. This mission is repeated 10 times with the same home position and waypoint. From Table 1 it can be seen that successfully landing with an average landing position error is 285.4 cm from the landing pad. The minimum position error is 26 cm, and the maximum position error is more than 5 meters.

C. Landing Using Computer Vision

Drone landing is done in two parts. The first part is testing by sending the drone manually close to the landing pad, then the landing switch is activated. The second part is done by combining the computer-based landing vision system with the waypoint. In the first test, a total of 10 trials were conducted. The average successful error difference is 46.4 cm. In this test the drone successfully landed with a minimum position error of 21cm from the center (see table 2). Figure 10 is the trajectory of the drone during the first part of the automatic landing. It can be seen in the figure, a large number of points corresponds to the process of drone centering position, then to the gradual descending of the drone, until it finally reaches the landing pad ground level.

Table 2
Landing using computer vision

Trial	Success	Error Position
1	Yes	120 cm
2	Yes	96 cm
3	Yes	37 cm
4	No	-
5	Yes	20 cm
6	Yes	30 cm
7	Yes	21 cm
8	Yes	25 cm
9	Yes	43 cm
10	Yes	26 cm

In the second part, waypoints are used to send the drone autonomously to hover without landing. The drone is then commanded to return to home (RTH) with the landing pad that should be detected by the camera sensor automatically. From the tests that have been carried out the drone has landed with a smaller error compared to landing using GPS.

The drone managed to detect and land with the smallest error of 21 cm (see table 3). With an error the average landing

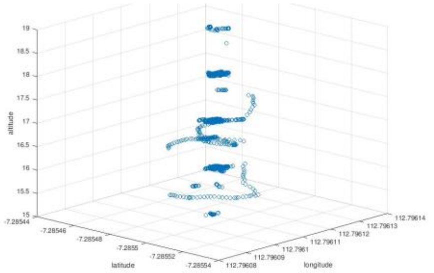

Figure 10. Automatic drone landing process trajectory.

that is successful is 45 cm. In a failed trial the drone still landed but did not rely on image processing. This is because RTH from the drone is too far from the landing pad, even though the drone flies the higher the landing pad remains undetected. So that the drone landed automatically after not detecting it for a specified time of 5 seconds. Figure 11 is the trajectory of the drone landing process when making an automatic landing. From Figure 11 we can see the drone flying towards the waypoint, then flying position hold and making an automatic landing.

Table 3
Trial of landing integration of waypoint / computer vision

Trial	Success	Error position
1	Yes	137 cm
2	Yes	24 cm
3	No	-
4	Yes	21 cm
5	No	-
6	Yes	53 cm
7	Yes	31 cm
8	Yes	48 cm
9	Yes	21 cm
10	Yes	25 cm

D. Precision Landing

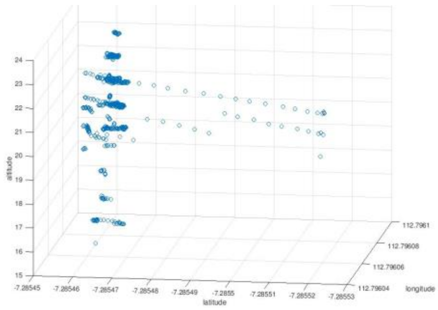

Figure 11. Automatic landing with a waypoint trajectory

Precision landing is not defined by default with certain numbers. After conducting a literature study it can be concluded that, precision landing is a landing by relying on vision algorithms to improve the accuracy of the landing. By relying on vision, the drone can find out the position of the landing pad. The smaller the landing pad, the better the level of landing precision. Based on some of the literature that has been carried out on precision landing studies, it can be concluded that the precision landing in general has a much better error than landing by GPS only. In this work, GPS-based landing test results show an average position error of 2,854 m. Using computer vision algorithms, from the results of the experiments in this work, the drone successfully made a precision landing with an average position error of 45 cm with a minimum landing position of 21 cm. Table 4 is a comparison of research on precision recording on drones.

Table 4
Resume precision landing

No	Paper	Error position
1	Auto Takeoff and Precision Landing Using Integrated GPS/Optical Flow Solution[16]	Maximum 0.27 m
2	Precision landing using an adaptive fuzzy multi-sensor data fusionarchitecture[20]	Less than 0.1 m
3	An Autonomous UAV with an Optical Flow Sensor for Positioning and Navigation[18]	0.30 m
4	Drone Precision Landing using Computer Vision[19]	Maximum 0.05 m
5	Precision Landing of a Quadrotor UAV on a Moving Target Using Low-cost Sensors[5]	Average 0.38 m
6	Multirotor UAV sensor fusion for precision landing[21]	0.08 m
7	Visually-Guided Landing of an Unmanned Aerial Vehicle[17]	Average 0.42 m
8	Precision landing for Drone Based on Computer Vision (this work)	Average 0.45 m

V. CONCLUSION

A quadcopter implementing on-board computer vision and autonomous control for precision landing has been constructed and tested. Based on experiment result, it can be safely concluded that precision landing with the help of computer vision gives a better landing accuracy on flight mode position hold. From experiment results, the drone is able to perform landing using computer vision with an average position error of 0.46 m and the minimum error of 0.21 m. This research also conducts landing experiment with GPS navigation, with average error of 2.854 m and minimum position error of 0.21m, notably higher than the results from visually-assisted precision landing.

REFERENCES

[1]C. N. Hamdani, R. E. A. K, and E. Iskandar, "Perancangan Autonomous Landing pada Quadcopter Menggunakan Behavior - Based Intelligent Fuzzy Control," *Jurnal Teknik ITS*, vol. 2, no. 2, 2013.

[2]W. Bai, F. Pan, B. Xing, C. Pan, and M. Pei, "Visual Landing System Of UAV Based On ADRC," *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 7509–7514, 2017.

[3]T. K. Venugopalan, "Autonomous Landing of an Unmanned Aerial Vehicle on an Autonomous Marine Vehicle," 2012 Oceans, 2012.

[4]R. Kharisman, K. Astrowulan, and E. Ak, "Perancangan Sistem Navigasi Menggunakan Kamera pada Quadcopter untuk Estimasi Posisi dengan Metode Neural Network," *Jurnal Teknik ITS*, vol. 3, no. 1, pp. 1–6, 2014.

[5]K. Ling, "Precision Landing of a Quadrotor UAV on a Moving Target Using Low-cost Sensors," University of Waterloo, Ontario, Canada.

[6]A. Yu, "Drones for Deliveries." https://scet.berkeley.edu/wp-content/uploads/ConnCarProjectReport-1.pdf, accessed 15 Jul. 19

[7]L. Jinwo and J. Hosang, "Drone Delivery Scheduling Simulations Focusing on Charging Speed, Weight And Battery Capacity :Case of Remote Islands in South Korea," in *Proceedings of the 2017 winter simulation conference*, 2017, pp. 4220–4227.

[8]A. W. Sudbury and E. B. Hutchinson, "A COST ANALYSIS OF AMAZON PRIME AIR (DRONE DELIVERY)," *Journal for Economic Educators*, Middle Tennessee State University, Business and Economic Research Center, vol. 16(1), pages 1-12, 2016.

[9]D. Domingos, G. Camargo, and F. Gomide, "Autonomous Fuzzy Control and Navigation of Quadcopters," *IFAC-Pap.*, vol. 49, no. 5, pp. 73–78, 2016.

[10]R. Mahtani and A. Ollero, "Control and Stability Analysis of quadcopter," in *International Conference on Computing, Mathematics and Engineering Technologies – iCoMET*, 2018, pp. 2–7.

[11]S. Huh and D. Hyunchul, "A Vision-Based Automatic Landing Method for Fixed-Wing UAVs," *Journal of Intelligent and Robotic Systems*, pp. 217–231, 2010.

[12]M. F. Sani and G. Karimian, "Automatic Navigation and Landing of an Indoor AR . Drone Quadrotor Using ArUco Marker and Inertial Sensors," *2017 International Conference on Computer and Drone Applications (IConDA)*, pp. 102–107, 2017.

[13]D. Honegger, L. Meier, P. Tanskanen, M. Pollefeys, and Z. Eth, "An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications." *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013, pp. 1736-1741.

[14]S. K. Phang, J. J. Ong, R. T. C. Yeo, B. M. Chen, and T. H. Lee, "Autonomous Mini-UAV for Indoor Flight with Embedded On-board Vision Processing as Navigation System," pp. 722–727, 2010.

[15]M. Engineering, "Lidar-guided Autonomous Landing of an Aerial Vehicle on a Ground Vehicle," pp. 228–231, 2017.

[16]M. K. S. Al-Sharman, "AUTO TAKEOFF AND PRECISION LANDING USING INTEGRATED GPS/INS/OPTICAL FLOW SOLUTION."

[17]S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Robot. Autom.*, vol. 19, no. 3, pp. 371–380, Jun. 2003.

[18]N. Gageik, M. Strohmeier, and S. Montenegro, "An Autonomous UAV with an Optical Flow Sensor for Positioning and Navigation," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 10, p. 341, Oct. 2013.

[19]L. Goeller, "Drone Precision Landing using Computer Vision." , Distributed Computing Group, Computer Engineering and Networks Laboratory, ETH Z ̈urich, https://pub.tik.ee.ethz.ch/students/2018-FS/SA-2018-21.pdf accessed 15 July 2019.

[20]M. K. Al-Sharman, B. J. Emran, M. A. Jaradat, H. Najjaran, R. Al-Husari, and Y. Zweiri, "Precision landing using an adaptive fuzzy multi-sensor data fusion architecture," *Appl. Soft Comput.*, vol. 69, pp. 149–164, Aug. 2018.

[21]M. Grzes, M. Slowik, and Z. Gosiewski, "Multirotor UAV sensor fusion for precision landing," *Aircr. Eng. Aerosp. Technol.*, vol. 91, no. 2, pp. 241–248, Feb. 2019.

[22]"BAB II Unmanned Aerial Vehicle (UAV)." [Online]. Available: http://digilib.unila.ac.id/5709/10/BAB II.pdf. [Accessed: 03-Jul-2019].

[23]A. Kaehler and G. Bradski, *Learning OpenCV 3 COMPUTER VISION IN C++ WITH THE OPENCV LIBRARY*, First Edit. United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. O'Reilly, 2016.

[24]M. Syuhada, "Realisasi pengenalan plat nomor kendaraan dengan metode histogram citra dan jaringan syaraf tiruan backpropagation," Fakultas Teknik, Universitas Lampung, 2015.

[25]A. M. Utama, D. Purwanto, and R. Mardiyanto, "Rancang Bangun Sistem Pengukuran Posisi Target dengan Kamera Stereo untuk Pengarah Senjata Otomatis," *J. Tek. ITS*, vol. 5, no. 2, pp. 216–220, 2016.

[26]P. Joshi, *OpenCV with Python By Example*. Birmingham: Packt Publishing Ltd., 2015.

[27]K. Nikolskaia, N. Ezhova, and A. Sinkov, "Skin Detection Technique Based on HSV Color Model and SLIC Segmentation Method," *Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists*, At Yekaterinburg, Russia, vol. 211, 2018.

[28]B. Y. Budi Putranto, W. Hapsari, and K. Wijana, "Segmentasi Warna Citra Dengan Deteksi Warna Hsv Untuk Mendeteksi Objek," *J. Inform.*, vol. 6, no. 2, 2011.

[29]M. Computing and L. Alsadoon, "A Hybrid Communication System Using Pulse Position and Width Modulation," *Int. J. Comput. Sci. Mob. Comput.*, vol. 8, no. 3, pp. 1–13, 2019.

[30] Neha R. Laddha, A.P.Thakare, "A Review on Serial Communication by UART," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 1, pp. 366–369, 2013.

[31]V. Flori, "OMNIBUS F4 V3 – Documentation," vol. 64. 2017.

[32]R. P. M. B, "Raspberry Pi 3 Model B + datasheet.".

[33]Arduino CC, "Arduino Nano," *Arduino Nano R3*. https://store.arduino.cc/usa/arduino-nano [accessed 15 July 2019]